

过滤数据

一、 where

数据库表一般包含大量数据，很少需要检索所有的行。只检索所需要的数据需要指定搜索条件，搜索条件也称为过滤条件。

where子句在表名之后出现。

```
select prod_name ,prod_price from products where prod_price = 3.49;
```

代表只返回价格等于3.49的数据。

有的时候浮点数数据显示的结果可能小数点后面的0的个数不一样，这往往是因为DBMS指定了所用的数据类型及其默认行为。

过滤

数据也可以在应用层过滤，为此SQL的select语句为客户端应用检索出超过实际所需的数据，然后客户端代码对返回数据进行循环，提取所需要的行。

where子句的位置：

同时使用order by和where子句的时候，应该让order by位于where之后，否则会产生错误。

1.1 where子句的操作符

= <> != < <= != < > >= != > BETWEEN IS NULL

1.2 不匹配

<>

```
select vend_id , prod_name from products where vend_id <> '';
```

何时使用引号呢？单引号用来限定字符串，如果将值和字符串比较，就需要限定引号。用来与数值列进行比较的值不用引号。

相同的例子如下：

```
select vent_id , prod_name from products where vend_id != '123';
```

!= 和<>通常可以互换。并非所有的DBMS系统都支持所有的操作符。

1.3 范围检查

BETWEEN 需要指定低端值和高端值。

BETWEEN A AND B

1.4 空值检查

在一个列不包含值的时候，称之为空值。NULL

NULL: 无值，它和字段0、字符串空、或者仅仅包含空格不同。

确定值是否为NULL,不能简单使用=NULL，select语句有一个特殊的where子句，用来检查是否具有null值。

```
select prod_name from products where prod_price is null;
```

二、高级过滤

组合WHERE子句，SQL允许多个where子句，这些子句有两种方式使用，and或者or。

操作符: 联结或者改变where子句中的关键字，也称为逻辑操作符。

2.1 AND

AND指示DBMS返回所有给定条件的行。

2.2 OR

OR和AND相反，指示DBMS匹配任意条件。许多DBMS通常在OR的条件中，前面的满足之后就不在计算后面的条件了。

where子句中使用的关键字，用来表示检索匹配任一给定的条件。

2.3 求值顺序

where子句中可以包含任意数目的AND和OR，允许结合进行复杂高级的过滤。

SQL优先处理AND，为了避免优先级顺序差异造成困扰，建议使用小括号来明确表示优先级。小括号比AND或者OR有更高的优先级顺序。

2.4 in

功能与or相当，in后面跟由逗号分隔的合法值，这些值包含在括号中。

优点：

- 有很多合法值的情况下，in更清楚、明了
- 在和其他的and和or结合的时候，使用in更容易管理顺序
- in比一组or执行的更快
- in的最大优点，是可以包含其他的select语句，可以动态创立where条件。

2.5 not

not只有一个作用，就是否定它后面的任何条件。它可以用在要过滤的列之前也可以用在之后。

```
select prod_name from products where not vend_id = "123" order by prod_name;
```

否定vend_id的值，不能等于123；

它与下面的sql等效：

```
select prod_name from products where vend_id <> "123" order by prod_name;
```

当然这种简单的sql看不出not的优势，在更复杂的子句中，not非常有用。

三、通配符过滤

利用通配符可以创建比较特定数据的搜索模式。

通配符：用来匹配值的一部分的特殊字符。

搜索模式：由字面值、通配符或者两者组合成的搜索条件。

它本身是普通的sql的where子句中包含的条件，只不过这些条件中含有特殊的意义。

为了使用通配符，则必须使用like操作符，它指示DBMS，跟在后面的搜索模式使用通配符来解决而不是简单地相等匹配。

谓词: 操作符一般被称为操作符，那它什么时候不是操作符呢？作为谓词的时候。精确地说like是谓词而不是操作符，虽然结果相同，但理解起来可能不同。

通配符的搜索，只能用于文本字段，非文本字段不支持通配符。

3.1 %

%匹配任何字符出现任何次数。

```
select prod_id ,prod_name from products where prod_name like 'Fish%';
```

则会匹配prod_name以Fish开头的行，Fish开头代表可以只有Fish，也可以在Fish后面跟随任意多个其他的字符。

异同: 在Access中，通配符使用*而不是%

区分大小写：搜索有时候可以区分大小写，如区分大小写，则注意过滤字段。

NULL: like '%'不会匹配到null的行

3.2 _下划线

用途和%类似，但是只匹配单个字符，而不是多个。

DB2不支持_

ACCESS使用?而不是_

3.3 方括号[]

[]指定字符集，必须匹配指定位置的一个字符。

不是所有的DBMS系统都支持集合[],ACCESS SQLServer是支持的。

如：

```
select cus_concat from customers where cust_concat like '[JM]%' order by cus_concat;
```

它只能匹配一个字符，任何多于一个的都匹配不上。[JM]之后跟随的%匹配一个字符之后的任何数目的字符，返回所需结果。

结果：返回cus_concat字段以J或者M开头的。

可以用脱字号来否定, ^conditionChar, 表示不在这些字符集中的一个字符。

如:

```
select cus_concat from customers where cust_concat like '[^JM]%' order by cus_concat;
```

结果: 返回cus_concat字段不以J也不以M开头的。

Access中的否定用的是!而不是^

当然也可以直接用NOT来否定, ^的唯一优点是在使用多个where子句的时候可以简化语法。

如:

```
select cus_concat from customers where not cus_concat like '[JM]%' ORDER BY cust_concat;
```

3.4 通配符的技巧

通配符一般比之前的简单检索要耗时。因此有以下技巧:

- 不要过度使用, 如果其他的简单方式可以实现, 就不要使用通配符
- 在确实需要的时候使用, 也尽量不要用在搜索模式的开始。把通配符用到最开始, 则搜索起来是最慢的。
- 仔细注意通配符的位置, 如果放错地方可能会适得其反。